

# Software–Projekt 2 2013/2014

VAK 03-BA-901.02

## Projektplan

Sebastian Bredehöft	sbrede@tzi.de	2751589
Patrick Damrow	damsen@tzi.de	2056170
Tobias Dellert	tode@tzi.de	2936941
Tim Ellhoff	tellhoff@tzi.de	2520913
Daniel Pupat	dpupat@tzi.de	2703053
Mohamadreza (Amir) Khostevan	amirkh@tzi.de	1234567

## Contents

<b>1</b>	<b>Einleitung</b>	<b>4</b>
1.1	Projektübersicht . . . . .	4
1.1.1	Ziele . . . . .	4
1.1.2	Hauptarbeitsaktivitäten und -produkte 6	
1.1.3	Haupt-Meilensteine und grober Zeitplan . . . . .	6
1.1.4	Benötigte Ressourcen . . . . .	7
1.1.5	Budget . . . . .	7
1.1.6	Kontaktdaten des Kunden . . . . .	8
1.1.7	Mitarbeiter . . . . .	8
1.2	Auszuliefernde Produkte . . . . .	8
1.3	Evolution des Plans . . . . .	8
1.4	Referenzen . . . . .	8
1.5	Definitionen und Akronyme . . . . .	8
<b>2</b>	<b>Projektorganisation</b>	<b>9</b>
2.1	Prozessmodell . . . . .	9
2.2	Organisationsstruktur . . . . .	9
2.3	Organisationsgrenzen und -schnittstellen . . . . .	9
2.4	Verantwortlichkeiten . . . . .	9
<b>3</b>	<b>Managementprozess</b>	<b>9</b>
3.1	Managementprozess und -prioritäten . . . . .	9
3.2	Annahmen, Abhängigkeiten und Einschränkungen . . . . .	9
3.3	Risikomanagement . . . . .	9
3.4	Projektüberwachung . . . . .	10
3.5	Mitarbeiter . . . . .	10
<b>4</b>	<b>Technische Prozesse</b>	<b>10</b>
4.1	Methoden, Werkzeuge und Techniken . . . . .	10
4.1.1	Entwicklungsplattform . . . . .	10
4.1.2	Entwicklungsmethode . . . . .	11
4.1.3	Programmiersprache und Bibliotheken . . . . .	11
4.2	Dokumentationsplan . . . . .	11
4.2.1	Codingstyle . . . . .	11
4.2.2	Kommentarsprache . . . . .	11
4.2.3	JavaDoc . . . . .	11
4.2.4	Begleitende Dokumentation . . . . .	12
4.3	Unterstützende Projektfunktionen . . . . .	12
<b>5</b>	<b>Arbeitspakete, Zeitplan und Budget</b>	<b>12</b>
5.1	Arbeitspakete . . . . .	12

5.2	Zeitplan und Abhängigkeiten . . . . .	13
5.3	Ressourcenanforderung . . . . .	13
<b>6</b>	<b>Sonstige Elemente</b>	<b>14</b>
6.1	Pläne für die Konvertierung von Daten . . . . .	14
6.2	Managementpläne für Unterauftragsnehmer . . . . .	14
6.3	Ausbildungspläne . . . . .	14
6.4	Raumpläne . . . . .	14
6.5	Installationspläne . . . . .	14
6.6	Pläne für die Übergabe des Systems . . . . .	14

## Version und Änderungsgeschichte

*Die aktuelle Versionsnummer des Dokumentes sollte eindeutig und gut zu identifizieren sein, hier und optimalerweise auf dem Titelblatt.*

Version	Datum	Änderungen
0.1	14.10.2013	Ziele hinzugefügt.
0.1.1	14.10.2013	Ziele vervollständigt
0.2	15.10.2013	Hauptarbeitsaktivitäten und -produkte hinzugefügt.
0.3	15.10.2013	Meilensteine eingefügt.
0.4	15.10.2013	Benötigte Ressourcen -Menschen hinzugefügt.
0.5	16.10.2013	Ressourcen ergänzt -Hardware und -Räume.
1.0	20.10.2013	Erste veröffentlichte Version.

## 1 Einleitung

### 1.1 Projektübersicht

#### 1.1.1 Ziele

Das Ziel unserer Gruppe IT\_R3V0LUT10N ist es, das Softwareprojekt 2 der Universität Bremen zu bestehen. Dies setzt die Einhaltung der Fristen und Termine, eine ausreichende Fertigstellung des Projekts und die Abgabe aller in SWP2 geforderten Dokumente wie Projektplan, Anforderungsspezifikation und Angebot, Architekturbeschreibung, Schnittstellenbeschreibung, Testplan inklusive Blackbox-Tests und ein elektronisch geführtes Berichtsheft voraus. Darüber hinaus wollen wir einen GUI-Prototypen erstellen und den Akzeptanztest bestehen. Ein Bibliothekssystem zu erstellen steht aber im Vordergrund.

Das Bibliothekssystem beinhaltet sowohl eine Website, als auch einen Zugang für mobile Geräte mit kleinem Display. Ziel ist es, die Mindestanforderungen<sup>1</sup> und eventuell weitergehende Funktionen zu implementieren.

Zu den Mindestanforderungen gehören die Erstellung und Abgabe einer Bibliothekssoftware, eines Serverprogramms mit Datenbankanbindung, einen Administrationszugang und einen Zugang für mobile Geräte mit kleinem Display. Wir haben uns entschieden den Zugang für die mobilen Geräte in Form einer Android-App zu realisieren, weil wir dies für Zeitgemäß und einfacher für den Leser, an den diese Form des Zugangs gerichtet ist, halten. Die zu erstellende Bibliothekssoftware dient in erster Linie zur Verwaltung des Medienbestandes der Bibliothek und dem Verleihen dieser Medien an der Oberschule Rockwinkel. Der Administrationszugang wird benötigt um Bibliothekare anzulegen, zu löschen, deren Stammdaten zu ändern, oder deren Rechte innerhalb der Software zu verändern.

---

<sup>1</sup>[http://www.informatik.uni-bremen.de/st/Lehre/swpII\\_1314/mindestanforderungen.html](http://www.informatik.uni-bremen.de/st/Lehre/swpII_1314/mindestanforderungen.html)

**(TODO: Weitere Features sind den oben genannten Mindestanforderungen zu entnehmen, oder werden im weiteren Verlauf des Projektplans erläutert.)**

### 1.1.2 Hauptarbeitsaktivitäten und -produkte

(TODO: Beschreibung)

Table 1: Hauptaktivitäten und -produkte

Aktivität	Arbeitsprodukt
Projektplanung	Projektplan
Anforderungsanalyse, erstellen des Angebots	Anforderungsspezifikation, Angebot
Globale Analyse, Konzeptionelles Modell, Modusblickwinkel, Ausführungsblickwinkel, Codeblickwinkel	Architekturbeschreibung
erstellen des Testplans	Testplan, Schnittstellentests
Implementierung	lauffähiges Programm

### 1.1.3 Haupt-Meilensteine und grober Zeitplan

#### M1 - 20.10.2013 Abgabe initialer Projektplan

Jedes Mitglied muss seinen Teil fertig gestellt haben. Anschließend werden alle Einzelteile zusammengeführt und von allen auf Korrektheit geprüft.

#### M2 - 13.11.2013 Anforderungsspezifikation (Intern)

Jedes Mitglied hat seinen Teil der Anforderungsspezifikation fertiggestellt. Anschließend werden die Teile zusammengeführt und von allen auf Korrektheit geprüft.

#### M3 - 17.11.2013 Abgabe der Anforderungsspezifikation

Meilenstein 2 muss bereits fertig sein. Abgabe der Anforderungsspezifikation via MEMS.

#### M4 - 18.12.2013 Architektur- und Schnittstellenbeschreibung, Testplan, Tests (Intern)

Jedes Mitglied muss seine Aufgaben erfüllt haben. Teile werden zusammengeführt und kontrolliert. Tests müssen implementiert sein.

#### M5 - 22.12.2013 Architekturbeschreibung, Testplan und Schnittstellentests fertig

Meilenstein 4 muss bereits erreicht worden sein. Tests wurden lauffähig implementiert. Abgabe via MEMS.

#### M6 - 26.01.2014 Erste lauffähige Basisversion

Jedes Mitglied muss seine Arbeitspakete fertig gestellt haben. Das Team muss die oben genannten Mindestanforderungen (TODO: referenz auf fußnote), die mit einem \* markiert sind, implementiert haben

#### M7 - 23.02.2014 Vollständige Abgabe der Dokumente und der Software

Die Software muss lauffähig und vollständig implementiert sein,

Abgabe des Build-/Installationsskriptes

#### 1.1.4 Benötigte Ressourcen

- **Menschliche Ressourcen**

An Menschlichen Ressourcen stehen sechs Informatik Studenten der Universität Bremen zur Verfügung. Wir haben als durchschnittliche Arbeitszeit pro Woche und Person einen Aufwand von ca. 14,5 Stunden für das Projekt errechnet. Dieser Wert ergibt sich folgendermaßen:

Für das Modul Software Projekt 2 gibt es 9CP. 1CP entspricht 30 Semesterstunden.  $9 \times 30 = 270$  Stunden. Da wir 19 Wochen lang an dem Projekt arbeiten werden, ergibt sich ein aufgerundeter Wert von 14,5 Stunden pro Woche ( $270 / 19 = 14,21$ ). Unsere Kontaktdaten sind dem Punkt Mitarbeiter zu entnehmen. **(TODO: Referenz)**

- **Hardware**

Jedes unserer Mitglieder ist im Besitz, oder hat Zugriff, auf Computer, die folgenden Anforderungen und Verfügbarkeiten gerecht werden müssen:

- zum Anfertigen der Dokumente wird ein Textsatzprogramm benötigt (L<sup>A</sup>T<sub>E</sub>X wird bevorzugt).
- für die Entwicklung der Software müssen Java-Runtime, ein Texteditor und eine Entwicklungsumgebung mit Android-SDK installiert sein.
- Git wird zum gleichzeitigen Bearbeiten der Dokumente und zum Datenaustausch der Entwickler benötigt.

- **Räume**

Das Team wird sich während der gesamten Projektlaufzeit Montags, soweit verfügbar, in einer der Lerninseln im GW2 A2370 oder A3440 der Universität Bremen von 10 Uhr bis 14 Uhr treffen. Weitere spezielle Räumlichkeiten werden nicht benötigt, da wir den Kontakt regelmäßig via Skype oder Email gewährleisten.

#### 1.1.5 Budget

Ein Budget für dieses Projekt in Form von Geld entfällt, da die Software im Rahmen des Moduls Software Projekt 2 entwickelt wird. Wenn wir über 19 Wochen (vom 14.10.2013 bis zum 23.02.2014) an dem Projekt mit 6 Studenten 14,5 Stunden pro Woche arbeiten, ergibt sich eine Gesamtsumme von 1653 Entwicklerstunden ( $19 \times 6 \times 14,5 = 1653$ ).

Wir entnehmen einer Studie von Gulp <sup>2</sup> das zwei Drittel der Software-Entwickler zwischen 60 und 80 Euro fordern. Da wir alle Studenten sind und somit noch in der Aus-

---

<sup>2</sup><http://www.gulp.de/presse/pressemitteilungen/marktstudie-freiberufliche-software-entwickler-sind.html>

bildung, setzen wir den Studienlohn für jeden Entwickler bei 40 Euro an. Somit würden sich für den Arbeitsaufwand der Entwicklerstunden Kosten von insgesamt 66.120 Euro ergeben.

### 1.1.6 Kontaktdaten des Kunden

*Oberschule Rockwinkel*

*Uppe Angst 31*

*28355 Bremen*

*Telefon: 0421 - 361 16 627*

*Fax: 0421 - 361 16 637*

*E-Mail: 416@bildung.bremen.de*

### 1.1.7 Mitarbeiter

Name	Email	Foto
Bredehöft, Sebastian	sbrede@tzi.de	bild einfügen
Damrow, Patrick	damsen@tzi.de	bild einfügen
Dellert, Tobias	tode@tzi.de	bild einfügen
Pupat, Daniel	dpupat@informatik.uni-bremen.de	bild einfügen
Ellhoff, Tim	tellhoff@tzi.de	bild einfügen
Khostevan, Mohamadreza (Amir)	amirkh@tzi.de	bild einfügen

## 1.2 Auszuliefernde Produkte

## 1.3 Evolution des Plans

ENTFÄLLT *Wird der Plan verändert? Wann? Wie oft? Von wem? Wenn bereits Aktualisierungen vorgesehen sind, welche sind das? Möglicherweise betrifft das die Zeitplanung, die Risikobewertung, oder andere Teile des Plans. Gibt es möglicherweise auch unvorhergesehene Aktualisierungen?*

## 1.4 Referenzen

## 1.5 Definitionen und Akronyme

ENTFÄLLT *Hier sollen Begriffe definiert werden, die nötig sind, um den Projektplan zu verstehen. Diese kommen insbesondere aus der Welt des Kunden (Projektdomäne) und der Welt des Softwareproduzenten.*



## 2 Projektorganisation

ENTFÄLLT

### 2.1 Prozessmodell

ENTFÄLLT

### 2.2 Organisationsstruktur

ENTFÄLLT *Genaue Beschreibung der Rollen, Rechte und Pflichten!*

*z.B. auch regelmäßiges Treffen im Chat, Einrichtung einer Groupware oder eines Forums, o.ä. ...*

### 2.3 Organisationsgrenzen und –schnittstellen

ENTFÄLLT *Hierher gehören auch evtl. Kontaktpersonen für Fremdbibliotheken u.ä.*

### 2.4 Verantwortlichkeiten

ENTFÄLLT

## 3 Managementprozess

### 3.1 Managementprozess und –prioritäten

### 3.2 Annahmen, Abhängigkeiten und Einschränkungen

### 3.3 Risikomanagement

*Wenn Ihr Euch entschieden habt, bestimmte vorbeugende Maßnahmen durchzuführen, solltet Ihr dies deutlich kennzeichnen. Hoffentlich haben diese Maßnahmen dann einen Einfluss auf Eintrittswahrscheinlichkeit oder Schadenshöhe (zum Beispiel ist die Eintrittswahrscheinlichkeit von komplettem Datenverlust durch regelmäßige Backups deutlich geringer). Daher solltet Ihr für diese Fälle dann die verringerten Werte für Eintrittswahrscheinlichkeit, Schadenshöhe und Risikopotential zusätzlich angeben.*

*Wie werden neue Risiken erkannt/erfasst? Wer ist für was zuständig? Wie ist der Informationsfluss? ...*

*Dieser Teil ist ein wichtiger Schwerpunkt des Projektplans und sollte daher ausführlich behandelt werden.*

### 3.4 Projektüberwachung

*Wie wird der Projektstatus verfolgt? Wie stellt Ihr sicher, dass der Phasenleiter jederzeit über den Stand der Entwicklung informiert ist? Wie werden Probleme bzw. Verzögerungen frühzeitig erkannt und angegangen?*

### 3.5 Mitarbeiter

*Kompetenzen der und Anforderungen an die Mitarbeiter.*

## 4 Technische Prozesse

### 4.1 Methoden, Werkzeuge und Techniken

#### 4.1.1 Entwicklungsplattform

Folgende Werkzeuge werden im Entwicklungsprozess von uns benutzt:

- Eclipse<sup>3</sup> ist unsere Entwicklungsumgebung (beinhaltet AndroidSDK<sup>4</sup> für die Androidentwicklung)
- Maven<sup>5</sup> ist unser Build-Management Tool
- GlassFish 3.1<sup>6</sup> ist unser Application-Server
- JUnit<sup>7</sup> ist unser Framework zum testen
- GitHub<sup>8</sup> zur Versionsverwaltung
- GanttProject<sup>9</sup> für Gantt-Diagramme
- L<sup>A</sup>T<sub>E</sub>X<sup>10</sup> zur Dokumentenerstellung

**(TODO: Sind das alle?)**

---

<sup>3</sup><http://www.eclipse.org/>

<sup>4</sup><https://developer.android.com/sdk/index.html>

<sup>5</sup><http://maven.apache.org/>

<sup>6</sup><http://glassfish.java.net/>

<sup>7</sup><http://junit.org/>

<sup>8</sup><http://github.com/>

<sup>9</sup><http://www.ganttproject.biz/>

<sup>10</sup><http://www.latex-project.org/>

#### 4.1.2 Entwicklungsmethode

(TODO: Abhängigkeit von Prozessmodell, Mittwoch abstimmen)

#### 4.1.3 Programmiersprache und Bibliotheken

Die Programmiersprache wird Java (mindestens Version 5) sein. Außerdem wird in geringen Umfang HTML und XML benutzt.

Ob und welche Bibliotheken genutzt werden, kann zu diesem Zeitpunkt (*Abgabe: 20.10.2013*) nicht gesagt werden.

(TODO: Android SDK)

Sobald wir Bibliotheken nutzen, wird dieser Punkt aktualisiert.

### 4.2 Dokumentationsplan

Wir werden als Ergebnis verschiedene Dokumentationen vorweisen können. Diese sind:

- Nutzerhandbuch
- Installationsanleitung
- Dokumentation des Quellcodes

#### 4.2.1 Codingstyle

Unsere Implementierungen werden sich an die *Code Conventions for the Java Programming Language*<sup>11</sup> halten.

(TODO: \*.tex Dateien)

#### 4.2.2 Kommentarsprache

Die Sprache in der unsere Kommentare verfasst sind, wird Deutsch sein. Dies verhindert mögliche Missverständnisse innerhalb unserer Gruppe.

#### 4.2.3 JavaDoc

Wir benutzen JavaDoc zur Dokumentation unseres Quellcodes. Dieses lässt eine einfache Erstellung von HTML-Dokumentationsdateien zu. Zur zukünftigen Wartung wird bis auf triviale Codezeilen, der komplette Code in JavaDoc dokumentiert.

---

<sup>11</sup><http://www.oracle.com/technetwork/java/codeconv-138413.html>

#### 4.2.4 Begleitende Dokumentation

(TODO: Weitere Erläuterung zur Doku?)

### 4.3 Unterstützende Projektfunktionen

In Abschnitt (TODO: Abschnitt nennen) werden unsere Phasenleiter benannt, die für die jeweiligen Phasen verantwortlich sind und als Ansprechpartner und Leiter dienen.

Unsere Projektdateien stehen jederzeit auf *GitHub*<sup>12</sup> zur Verfügung. Außerdem werden regelmäßig von *jedem* Gruppenmitglied Datensicherungen, in Form von Updates und Backups des Repositorys auf dem eigenen Rechner stattfinden.

(TODO: Qualitätssicherung)

## 5 Arbeitspakete, Zeitplan und Budget

*Dieser Teil ist ein zweiter Schwerpunkt des Projektplans. Hier sollt Ihr die nächste Phase detailliert planen (siehe Arbeitspakete). Die weiteren Phasen sollen ebenfalls wenigstens grob geplant werden. Ein Gantt-Diagramm ist zwingend!*

*Ihr sollt den Plan in der kommenden Phase auch tatsächlich benutzen – und so Erfahrungen sammeln, was evtl. bei der Planung unberücksichtigt blieb. Bei der nächsten Zeitplanung (für die nächste Phase) bekommt Ihr dann evtl. eine noch bessere Planung hin.*

### 5.0.1 Anmerkungen und Annahmen

Wir haben bisher nur die Phasen Projektplan und Anforderungsspezifikation vollständig in den Arbeitspaketen und Zuteilungen behandelt. Die übrigen Phasen des Projekts - Architekturbeschreibung, Implementierung und Test - können zu diesem Zeitpunkt noch nicht detailliert beschrieben werden, sondern erfolgen stattdessen in grobem Format.

Arbeitspunkte zusammen

### 5.1 Arbeitspakete

Im Folgenden sind die einzelnen Arbeitspakete des Abschnitts Projektplan aufgeführt.

---

<sup>12</sup>asd

## 1. Arbeitspakete für Projektplan

### *Arbeitspaket 1.1*

Bezeichnung	Vorbereitung für Projektplanerstellung
Beschreibung:	
Hauptverantwortlicher	Prüfsumme
Abhängigkeit	....
Ressourcen	....
Aufwand und Gesamtdauer	10h, 120h
Beginn	....
Ende	....
Mindestanforderungen:	

### *Arbeitspaket 1.2*

Bezeichnung	Vorbereitung für Projektplanerstellung
Beschreibung:	
Hauptverantwortlicher	Prüfsumme
Abhängigkeit	...
Ressourcen	GitHub,...
Aufwand und Gesamtdauer	10h, 120h
Beginn	13.10.2013
Ende	20.10.2013
Mindestanforderungen:	

*Besonderen Wert legen wir auf die Granularität der APs. Diese sollten von 1-2 Personen in max. einer Woche Zeitdauer (kalendarisch, nicht Aufwand) bearbeitbar sein. Die Beschreibungen sollten so genau sein, dass der Bearbeiter damit genau weiß, was zu tun ist.*

## 5.2 Zeitplan und Abhängigkeiten

*Die Abhängigkeiten zwischen Arbeitspaketen oder Meilensteinen müssen genannt werden, sowie im Gantt-Diagramm eingezeichnet werden. Der kritische Pfad soll angegeben und/oder eingezeichnet werden!*

## 5.3 Ressourcenanforderung

*Jedem Arbeitspaket muss mind. ein Bearbeiter zugeordnet werden. Die Zuordnung der ganzen Gruppe sollte nur in Ausnahmefällen erfolgen – und dann vermutlich begründet werden!*

## **6 Sonstige Elemente**

ENTFÄLLT

### **6.1 Pläne für die Konvertierung von Daten**

ENTFÄLLT

### **6.2 Managementpläne für Unterauftragsnehmer**

ENTFÄLLT *Wenn Fremdbibliotheken benutzt werden...*

### **6.3 Ausbildungspläne**

ENTFÄLLT *Hierunter fallen z.B. auch interne Schulungen, die Ihr durchführen wollt.*

### **6.4 Raumpläne**

ENTFÄLLT ...

### **6.5 Installationspläne**

ENTFÄLLT ...

### **6.6 Pläne für die Übergabe des Systems**

ENTFÄLLT ...